TEST -1

1. Create one superclass HillStations and three subclasses Manali, Mussoorie, Gulmarg. Subclasses extend the superclass and override its location() and famousFor() method. i.call the location() and famousFor() method by the Parent class’, i.e. Hillstations class. As it refers to the base class object and the base class method overrides the superclass method; the base class method is invoked at runtime. ii.call the location() and famousFor() method by the all subclass’,and print accordingly.

SOLUTION:

```
class HillStations
{
    void location()
     {
       System.out.println("Located in the mountains.");
     }

    void famousFor()
     {
       System.out.println("Famous for natural beauty and scenic views.");
     }
}

class Manali extends HillStations {
    @Override
    void location()
     {
       System.out.println("Located in Himachal Pradesh, India.");
     }

    @Override
    void famousFor()
     {
       System.out.println("Famous for snow-capped mountains, adventure sports,
and Hadimba Temple.");
     }
}

class Mussoorie extends HillStations {
    @Override
    void location()
     {
        System.out.println("Located in Uttarakhand, India.");
     }

    @Override
    void famousFor()
     {
        System.out.println("Famous for scenic views, Kempty Falls, and Mall Road.");
     }
}

class Gulmarg extends HillStations {
    @Override
    void location()
     {
        System.out.println("Located in Jammu and Kashmir, India.");
     }

    @Override
    void famousFor()
     {
        System.out.println("Famous for skiing, Gondola ride, and lush meadows.");
     }
}

public class Main {
    public static void main(String[] args) {
        HillStations hillStation = new HillStations();
        hillStation.location();
        hillStation.famousFor();
        Manali manali = new Manali();
        Mussoorie mussoorie = new Mussoorie();
        Gulmarg gulmarg = new Gulmarg();
        manali.location();
        manali.famousFor();
       mussoorie.location();
        mussoorie.famousFor();
       gulmarg.location();
```

```
        gulmarg.famousFor();
    }
}
```

OUTPUT:

```
Located in the mountains.
Famous for natural beauty and scenic views.
Located in Himachal Pradesh, India.
Famous for snow-capped mountains, adventure sports, and Hadimba Temple.
Located in Uttarakhand...
```

2. Create abstract class vaccine.Create two variables age(int),nationality(String).create 2 concrete methods firstDose() and secondDose(). Scenario 1:user can take the first dose if the user is Indian and age is 18.After vaccination the user has to pay 250rs(which will be displayed on the console). Scenario 2: Users are eligible to take the second dose only after completing the first dose. Scenario 3: create abstract method boosterDose() in abstract class Vaccine.Create one implementation class vaccinationSuccessful, where implement boosterDose() method. Create main class vaccination and invoke all methods accordingly. [Hint:Create constructor to initialize variables age and nationality,Use flow control(Ifelse) to check condition]

SOLUTION:

```
abstract class Vaccine {
    protected int age;
    protected String nationality;
    public Vaccine(int age, String nationality) {
        this.age = age;
        this.nationality = nationality;
    }
    abstract void boosterDose();
    public void firstDose() {
        if (nationality.equalsIgnoreCase("Indian") && age >= 18) {
            System.out.println("You are eligible for the first dose.");
            System.out.println("Please pay 250 Rs.");
        } else {
            System.out.println("You are not eligible for the first dose.");
        }
    }
    public void secondDose() {
        System.out.println("You can take the second dose after completing the first dose.");
    }
}
class VaccinationSuccessful extends Vaccine {
    public VaccinationSuccessful(int age, String nationality) {
        super(age, nationality);
    }
    @Override
    void boosterDose() {
        System.out.println("Congratulations! You have received the booster dose.");
    }
}
public class Vaccination {
    public static void main(String[] args) {
        VaccinationSuccessful user1 = new VaccinationSuccessful(20, "Indian");
        user1.firstDose();
        user1.secondDose();
        user1.boosterDose();
    }
}
```

OUTPUT:

```
You are eligible for the first dose.

Please pay 250 Rs.

You can take the second dose after completing the first dose.

Congratulations! You have received the booster dose.
```

3. Write code to determine if the string is a palindrome.
   input string: Madam.

   SOLUTION:

```
public class PalindromeChecker {

    public static boolean isPalindrome(String text) {
        String processedText = text.replaceAll("[^a-zA-Z0-9]", "").toLowerCase();
        int left = 0;
        int right = processedText.length() - 1;
        while (left < right) {
            if (processedText.charAt(left) != processedText.charAt(right)) {
                return false;
            }
            left++;
            right--;
        }
        return true;
    }

    public static void main(String[] args) {
        String input = "Madam";
        if (isPalindrome(input)) {
            System.out.println(input + " is a palindrome.");
        } else {
            System.out.println(input + " is not a palindrome.");
        }
}
}
```

   OUTPUT:

```
Madam is a palindrome.
```

4. You need to find and print all the unique characters in a given string.
   input string: java.

   SOLUTION:

```
import java.util.HashSet;

public class UniqueCharactersFinder {
    public static void main(String[] args) {
        String input = "java";
        String uniqueChars = findUniqueCharacters(input);
        System.out.println("Unique characters in the input string: " + uniqueChars);
    }

    public static String findUniqueCharacters(String str) {
        HashSet<Character> uniqueSet = new HashSet<>();
        StringBuilder result = new StringBuilder();

        for (char ch : str.toCharArray()) {
            if (!uniqueSet.contains(ch)) {
                uniqueSet.add(ch);
                result.append(ch);
            }
        }

        return result.toString();
    }
}
```

   OUTPUT:

```
Unique characters in the input string: jv
```